**Noțiuni teoretice**

**- Tablouri -**

Obs. Fiecare tablou este un obiect, derivat din clasa de baza *System.Array*. Accesul la elemente se face prin intermediul indicilor care incep de la 0 la numarul de elemente-1; orice depasire a indicilor duce la aparitia unei exceptii: *System.OutOfRangeException*.

**Tablouri unidimensionale**

*Declarare*: **int [ ] sir**;

Declararea de mai sus nu duce la alocare de spatiu pentru memorarea sirului; *instantierea* se poate face de ex.: **sir = new int[10]**;

***Exemplu:*** - secventa de mai jos citeste un sir de numere intregi:

**int[] x;**

**int n;**

**Console.Write("Numarul de elemente:");**

**n = int.Parse(Console.ReadLine());**

**x = new int[n];**

**for (int i = 0; i < n; i++) // *se poate scrie si: for (int i = 0; i < x.Length; i++)***

***// Length-returneaza nr elementelor vectorului.***

**{**

**Console.Write("Dati un nr. intreg:" );**

**x[i] = int.Parse(Console.ReadLine());**

**}**

***Observatie:*** In acest context **n** si **x** nu se pot declara la un loc, adica declaratii de genul:

**int [ ] x, n;** *sau* **int n, [ ] x**;

Se pot face **initializari** ale valorilor continute intr-un tablou:

**int[ ] a = new int[ ] {3, 5, 1};**

sau intr-o forma mai scurta:

**int[ ] a = {3, 5, 1};**

**Tablouri bidimensionale regulate**

*Declarare*: **int [ , ] x;**

*Instantiere* de ex.l: **x = new int[ 2, 2 ];**

*Referirea la elementul* aflat pe linia i si coloana j se face cu **x[i, j]**.

La declararea tabloului se poate face si *initializare*:

**int [ , ] x =new int[ , ] { {1, 2}, {3, 4}};** *sau, mai pe scurt:* **int [ , ] x = {{1, 2}, {3, 4}};**

***Exemplu:*** - secventa de program retine intr-un tablou bidimensional tabla inmultirii pana la 10 și o afiș.

**int[,] tb = new int[11,11];**

**for (int i=0; i<=10; i++)**

**for (int j=0; j<=10; j++)**

**tb[i,j] = i\*j;**

**for (int i=0; i<=10; i++)**

**for (int j=0; j<=10; j++)**

**Console.WriteLine("{0}\*{1}={2}",i,j,tb[i,j]);**

**Console.ReadLine();**

**// sau se poate scrie si ca mai jos, folosind metoda GetLength care apelata cu parametrul 0 intoarce numarul de linii, respectiv apelata pentru 1 intoarce numarul de coloane:**

**int[,] tb = new int[11,11];**

**for (int i=0; i<tb.GetLength(0); i++)**

**for (int j=0; j<tb.GetLength(1); j++)**

**tb[i,j] = i\*j;**

**for (int i=0; i<tb.GetLength(0); i++)**

**for (int j=0; j<tb.GetLength(1); j++)**

**Console.WriteLine("{0}\*{1}={2}",i,j,tb[i,j]);**

**Console.ReadLine();**

**Fișa de lucru**

* **Aplicații de tip consolă care folosesc tablouri unidimensionale și bidimensionale**

Pb.1

Realizaţi o aplicaţie de tip consolăcare citeşte de la tastatură un număr natural **n** (**2<n<=15**) şi construieşte în memorie o matrice **A** cu **n** linii şi **n** coloane în care orice element aflat pe prima linie sau pe prima coloană are valoarea **1** şi oricare alt element **Aij** din matrice este egal cu suma a două elemente din matrice, primul aflat pe linia **i** şi pe coloana **j-1** iar cel de-al doilea pe coloana **j** şi pe linia **i-1**. Matricea va fi afişată pe ecran, linie cu linie, numerele de pe aceeaşi linie fiind separate prin câte un spaţiu.

**Exemplu:** pentru **n=4** , se obţine matricea:

1 1 1 1

1 2 3 4

1 3 6 10

1 4 10 20

Pb. 2

Realizaţi o aplicaţie de tip consolăcare construieşte în memorie o matrice cu **4** linii şi **3** coloane ale cărei elemente sunt numere întregi (cu maximum **3** cifre fiecare), citite de la tastatură, şi afişează pe ecran, suma tuturor elementelor situate pe conturul matricei determinat de prima şi ultima linie respectiv prima şi ultima coloană a acestei matrice.

Pb. 3

Realizaţi o aplicaţie de tip consolă care citeşte de la tastatură un număr natural **n** (**n<20**), construieşte în memorie şi afişează pe ecran, matricea cu **n** linii şi **n** coloane, în care se vor memora în ordinea strict crescătoare a valorii, pe linii şi coloane, primele **n2** numere naturale nenule, pare.

**Exemplu:** dacă pentru **n** se citeşte valoarea **4** se afişează:

2 4 6 8

10 12 14 16

18 20 22 24

26 28 30 32

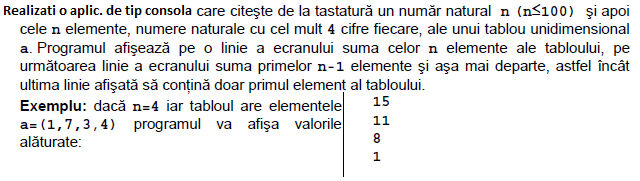
Pb. 4

Realizaţi o aplicaţie de tip consolă care citeşte de la tastatură un număr natural **n** (**n****50**) şi construieşte în memorie o matrice cu **n** linii şi **n** coloane, ale cărei elemente sunt numere întregi citite de la tastatură. Pentru fiecare coloană a matricei, în ordine, programul afişează pe ecran cel mai mic număr de pe respectiva coloană. Numerele afişate vor fi separate prin câte un spaţiu.

Pb. 5

Realizaţi o aplicaţie de tip consolă care citeşte de la tastatură două numere naturale nenule **m** şi **n** (**m≤10**, **n≤10**) şi cele **m\*n** elemente (numere naturale mai mici decât **100**) ale unui tablou bidimensional cu **m** linii, numerotate de la **1** la **m**, şi **n** coloane, numerotate de la **1** la **n**; programul construieşte în memorie şi afişează pe ecran tabloul după eliminarea liniei **1**.

Pb. 6

****